Multi-view Relighting using a Geometry-Aware Network

Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, George Drettakis

Multi-view Relighting using a Geometry-Aware Network

JULIEN PHILIP, Université Côte d'Azur and Inria
MICHAËL GHARBI, Adobe
TINGHUI ZHOU, UC Berkeley
ALEXEI A. EFROS, UC Berkeley
GEORGE DRETTAKIS, Université Côte d'Azur and Inria

(a) our algorithm can relight a single-illumination drone video dynamically to synthesize a "time-lapse" effect

(b) single-view input

(c) three relit outputs: here we built the *proxy* geometry using internet photos of the same location

Fig. 1. Two applications of our multi-view relighting system. (a) We show five different frames from a drone video (copyright Namyeska youtu.be/JHeDP7_YBos used with permission) relit with a "time-lapse" effect of a rotating sun (see supplemental for the full video). A user can also relight a photograph of a known landmark (b) to different target lighting conditions (c). For this, we applied our algorithm to a collection of 50 internet images of the same location .

We propose the first learning-based algorithm that can relight images in a plausible and controllable manner given multiple views of an outdoor scene. In particular, we introduce a *geometry-aware* neural network that utilizes multiple geometry cues (normal maps, specular direction, etc.) and source and target shadow masks computed from a noisy *proxy geometry* obtained by multi-view stereo. Our model is a three-stage pipeline: two sub-networks refine the source and target shadow masks, and a third performs the final relighting. Furthermore, we introduce a novel representation for the shadow masks, which we call *RGB shadow images*. They reproject the colors from all views into the shadowed pixels and enable our network to cope with inacuraccies in the proxy and the non-locality of the shadow casting interactions. Acquiring large-scale multi-view relighting datasets for real scenes is challenging, so we train our network on photorealistic synthetic data. At train time, we also compute a noisy stereo-based geometric proxy, this time from the synthetic renderings. This allows us to bridge the gap between the real and synthetic domains. Our model generalizes well to real scenes. It can alter the illumination of drone footage, image-based renderings, textured mesh reconstructions, and even internet photo collections.

Authors' addresses: Julien Philip, Université Côte d'Azur and Inria, julien.philip@inria.fr; Michaël Gharbi, Adobe, mgharbi@adobe.com; Tinghui Zhou, UC Berkeley, tinghuiz@eecs.berkeley.edu; Alexei A. Efros, UC Berkeley, efros@eecs.berkeley.edu; George Drettakis, Université Côte d'Azur and Inria, George.Drettakis@inria.fr.

0730-0301/2019/7-ART78 $15.00
https://doi.org/10.1145/3306346.3323013

CCS Concepts: • **Computing methodologies** → **Image manipulation**.

Additional Key Words and Phrases: Image relighting, Multi-view, Deep Learning

ACM Reference Format:
Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A. Efros, and George Drettakis. 2019. Multi-view Relighting using a Geometry-Aware Network. *ACM Trans. Graph.* 38, 4, Article 78 (July 2019), 14 pages. https://doi.org/10.1145/3306346.3323013

1 INTRODUCTION

Changing the illumination of an outdoor image is a notoriously difficult problem that requires the lighting to be modified consistently across the image, and shadows to be removed and resynthesized for the new sun position [Duchêne et al. 2015; Tchou et al. 2004; Yu et al. 1999]. Cast shadows are particularly challenging because an occluder can be arbitrarily far from the point it shadows, or even out of view.

The basic premise of our approach is to use multi-view information and approximate 3D geometry to reason about non-local lighting interactions and guide the relighting task. We introduce the first learning-based algorithm that can relight multi-view datasets of outdoor scenes (Fig. 1), which have become a commodity thanks to smartphone cameras, large-scale internet photo collections and drone cinematography. Our model uses a neural network designed to exploit geometric cues. It includes a careful treatment of cast shadows and is trained solely on realistic synthetic renderings.

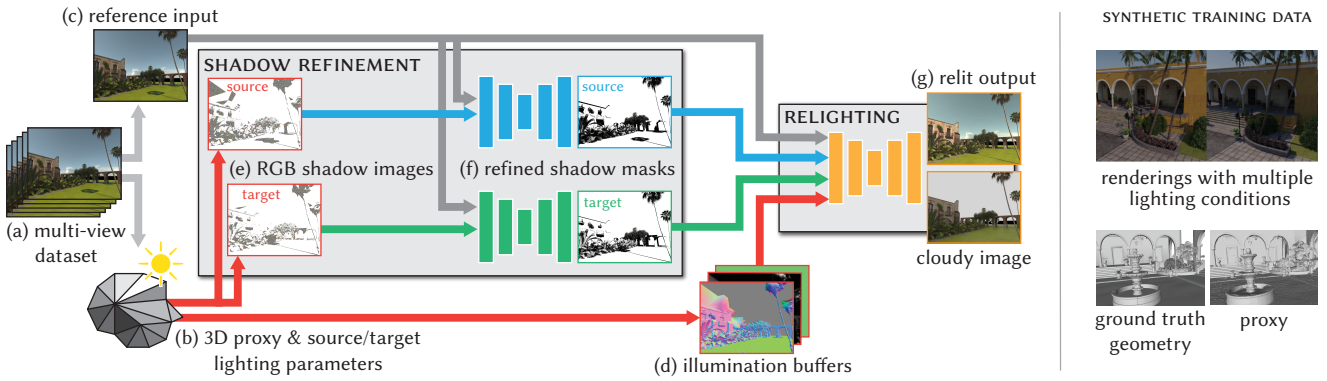


Fig. 2. Overview of our approach. **Left:** We use off-the-shelf stereo to create a 3D geometric proxy of the scene (b). The geometry is encoded as illumination buffers (d) and used to create RGB shadow images (e) that are independently refined by two networks (f), helping the final relighting network remove and re-synthesize shadows, and change the illumination (g) according to the desired novel lighting condition. **Right:** We train our model with synthetic data, including accurate ground truth geometry and renderings and an approximate proxy, created using synthetic renderings instead of photos. These two representations of the training scene allow the network to accurately refine shadows, enabling plausible relighting.

Our method has several applications: it allows automatic creation of a "time-lapse" effect by dynamically relighting drone videos (Fig. 1(a)). Or, if we only have a single photo, we can access online photos of the same place to relight the input photo (Fig. 1(b)). We can also relight images in traditional multi-view pipelines, e.g., Image-Based Rendering (IBR) or photogrammetry (Fig. 15).

Previous methods have difficulty with the type of input we target. Inverse-illumination methods [Loscos et al. 1999; Yu et al. 1999] cannot handle the approximate geometry of the proxy, while single-image relighting solutions struggle with cast shadows [Luan et al. 2017; Shih et al. 2013]. Finally, our solution significantly outperforms neural-network baselines (Sec. 5.2).

Method Overview. Given a set of images captured from multiple viewpoints (Fig. 2a), we start by building an approximate representation of the scene's geometry — a *proxy* — using off-the-shelf stereo [RealityCapture 2016; Snavely et al. 2006] (Fig. 2b). We can relight any *reference* view of that scene (Fig. 2c) — this could be one of the input images or a novel view obtained by IBR. The user provides a target illumination by specifying a sun direction vector and a scalar "cloudiness" level (or a sequence of such parameters for "time-lapse" effects). From the proxy, we then compute image-space *buffers* (Fig. 2d: normal maps, specular reflection direction, etc.) and shadow masks for the source and target illuminations. We perform relighting by training a neural network to map from the reference image, with extra buffers and shadow masks, to the novel lighting condition.

The importance of *accurate* shadow estimation for shadow removal has been previously demonstrated [Duchêne et al. 2015; Gryka et al. 2015; Guo et al. 2011]. But reconstruction errors in the proxy often lead to inacurrate masks a network cannot trust. This motivates our network design: we decompose our model into three sub-networks (Fig. 2). Two modules *refine* the source (resp. target) shadow masks (Fig. 2f) while the third implements the final relighting (Fig. 2g). The sub-networks are trained jointly but with different supervision: respectively ground truth shadow masks and ground truth relit images. Furthermore, instead of computing standard shadow masks from the proxy, we introduce *RGB shadow images* (Fig. 2e). These shadow images re-project colors from the shadow-casting geometry from all viewpoints into pixels in shadow, helping the network identify erroneously reconstructed shadow casters from the reprojected color (Fig. 4,5).

For supervised training, we need data corresponding to different lighting conditions of the exact same views, that is hard to capture with real photos. Instead, we use professionally-modeled, realistic synthetic scenes to generate physically-based renderings with many different viewpoints and lighting conditions. We introduce a flexible compositing methodology to generate a large variety of illuminations on-the-fly at training time. This avoids the combinatorial explosion in the number of images to render. Synthetic scenes also give us ground truth shadow masks.

To train the shadow refinement, it is impossible to capture real data and we cannot directly use the ground truth shadows cast by the synthetic geometry. A model trained with these perfectly accurate shadows would not generalize to real photographs, since it would have never seen the reconstruction errors of the stereo-based proxy. Instead, we generate an approximate 3D proxy for each synthetic training scene using stereo on renderings, from which we obtain the input illumination buffers and (inaccurate) RGB shadow images. The ground truth shadow masks are used as targets to supervise the refinement sub-networks. This approach makes our model robust to 3D reconstruction errors at test time and limits the generalization gap between real and synthetic data.

Contributions. In summary, we make the following contributions:

- An end-to-end learning method for multi-view relighting of outdoor scenes, guided by image-space buffers, namely shadow masks and illumination buffers, that are computed from a geometry proxy.
- A learning-based shadow refinement solution to remove and resynthesize shadows. It uses the input images as well as our

newly-introduced RGB shadow images to overcome reconstruction errors in the proxy.
- A training procedure that uses realistic synthetic scenes to flexibly generate multiple lighting conditions. Critically, we create a stereo-based proxy for each training scene which, together with the ground truth geometry, enables supervised learning for shadow refinement.

Although it is entirely trained on synthetic images, our algorithm generalizes to real multi-view datasets, and can modify the lighting in a much wider range of illumination conditions than previous methods (e.g., [Duchêne et al. 2015]). We evaluate our approach on real multi-view datasets, and show a variety of applications (Fig. 1,13,16).

2 RELATED WORK

Our method builds on several different areas. We first discuss traditional methods for single-image and multi-view relighting. One major challenge for relighting is the careful treatement of shadows. Our method removes and re-synthesizes shadows; we thus review the shadow removal literature. We also briefly review some aspects of image-to-image transformation research that is related to our solution.

2.1 Image-Based Relighting

Image-based relighting methods try to change the lighting conditions of an input image or a set of images. Early work ([Loscos et al. 1999; Marschner and Greenberg 1997; Yu et al. 1999], used laser scans or early user-assisted reconstruction algorithms to estimate geometry, and reflectance and/or environment lighting. Inverse global illumination is then used for relighting. More involved capture setups such as the Light Stage [Debevec et al. 2000; Wenger et al. 2005] allow for production-quality relighting, with wide-ranging applications in the film industry. In contrast, we target casual capture with a single camera (DSLR, phone or drone), providing approximate 3D geometry, which is most often unsuitable for inverse rendering methods.

Estimating the lighting environment in an image is an important step in relighting, with many proposed solutions (e.g., [Debevec 2002; Hold-Geoffroy et al. 2017; Lalonde et al. 2009a; Stumpfel et al. 2004]). Similarly, several reflectance estimation techniques have been proposed to assist relighting [Masselus et al. 2003, 2004]. Webcam sequences have also been used for relighting [Lalonde et al. 2009b; Sunkavalli et al. 2007], although cast shadows often require manual layering. Alternatively, online digital terrain and urban models registered to images can be used for approximate relighting [Kopf et al. 2008]. None of these methods satisfies all our requirements, i.e., plausible multi-view relighting including cast shadows for outdoors scenes using casual capture.

Another widely developed area of image relighting focuses on images of faces (e.g., [Peers et al. 2007; Wang et al. 2009; Wen et al. 2003]). The specific nature of face geometry and reflectance result in solutions that are not well adapted to the outdoors scenes we target.

Some methods target realistic object editing or compositing in single images [Karsch et al. 2011; Kholgade et al. 2014]. These methods give good results, but they do not adress major lighting changes, like editing cast shadows. They also require significant effort from the user to annotate the scene.

Several methods on multi-view image relighting have been developed, both for the case of multiple images sharing single lighting conditions [Duchêne et al. 2015], and for images of the same location with multiple lighting conditions (typically from internet photo collections) [Laffont et al. 2012; Xu et al. 2018]. For the single lighting condition, Duchêne et al. [2015], first perform shadow classification and intrinsic decomposition using separate optimization steps. Despite impressive results, artifacts remain especially around shadow boundaries and the relighting method fails beyond limited shadow motion. Our learning solution avoids the pitfalls of these optimization methods, and allows much larger sun motion (Section 5.3) as well as treating video sequences.

2.2 Intrinsic images, shadow estimation and removal

Intrinsic image decomposition and shadow removal methods are closely related to relighting. The classic Retinex work [Land and McCann 1971] inspired the intrinsic decomposition method of Weiss [2001], which used time-lapse sequences to compute shadow-free reflectance images. Many previous methods exist to explicitly detect and remove shadows, both in graphics and computer vision. See Sanin et al. [2012] for a survey. Most such methods operate on a single image, for example the work of Finlayson et al. [2006], which works well on shadows of relatively simple isolated objects. Other approaches include Lalonde et al. [2010] which uses Conditional Random Fields to detect the shadow, or Mohan et al. [2007] which is a gradient-based solution for shadow removal. These methods typically do not address relighting, which is our main goal. User assisted methods have also been developed [Shor and Lischinski 2008; Wu et al. 2007] but our automated approach is more practical for multi-view datasets.

Even before the massive adoption of deep CNNs, learning methods were proposed to remove shadows from images. The method of Guo et al. [2011], detects pairs of points in shadow/light using a learning approach, and subsequently removes shadows with an optimization. More recently, deep learning has been used for shadow removal [Qu et al. 2017], using pretrained features, global and local information. Generative Adversarial Networks (GANs) have also been used for shadow detection and removal, e.g., conditional GANs [Wang et al. 2018], where a first GAN learns to generate the shadow mask, which is then used by the second network to remove shadows. As with previous shadow removal methods, relighting is not addressed in this work. Recent deep learning methods achieve good results for shadow removal, but most often do not address *moving* shadows (especially cast shadows) and changing the overall lighting conditions. Handling such changes in lighting is a much more complex problem; our solution uses geometry and synthetic training data, achieving plausible relighting with cast shadows. We provide comparisons with baseline methods using such solutions in Section 5.2.

2.3 Deep learning for image-to-image transformations

The *Pix2Pix* method [Isola et al. 2017] uses a U-net [Ronneberger et al. 2015] to perform many different image transformation tasks with remarkable success, even though the quantity of training data is quite low compared to other methods. Similarly, ResNet-like architectures [He et al. 2016] have been particularly successful in large image transformation tasks [Zhu et al. 2017], thanks to the residual blocks that preserve useful information in the network. There has been a body of work on transforming images, including day-to-night [Liu et al. 2017] changes. While impressive, the results of these methods typically generated by GANs are lacking in consistency and ease of control. Finally, there has also been work on face or body relighting using deep learning (e.g., [Kanamori and Endo 2018; Shu et al. 2017]); as with older methods, the specific technical choices for faces or bodies result in methods that are not necessarily adapted for relighting of outdoor scenes, especially since the extent of outdoors scenes results in much more non-local effects.

3 GEOMETRY-AWARE RELIGHTING NETWORK

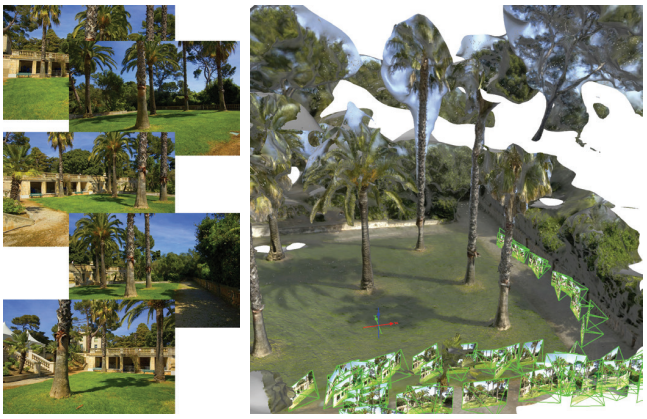
Our relighting solution is built around a neural network that takes one image from a multi-view dataset , and a set of corresponding image-space buffers as input, and produces a new image, with the lighting altered. We identified three key difficulties to successfully implement this image transformation: modeling the *illumination* changes (color, intensity, etc), and *removing* and *resynthesizing* cast shadows.

To overcome these difficulties, our learning solution exploits a *geometric 3D proxy* which we obtain by first calibrating the input virtual cameras using structure from motion (SfM) [Snavely et al. 2006], then running a Multi-View Stereo algorithm [Goesele et al. 2007; RealityCapture 2016]. Fig. 3 illustrates this procedure.

Because our CNN operates in the image domain, we encode the geometry and lighting parameters as image-space *illumination buffers B*. These include normal maps, per-pixel specular reflection direction, etc. See Section 3.3. In our ablation study, we found these buffers to be instrumental in synthesizing plausible novel illuminations (Section 5.5).

Furthermore, the proxy gives us a particularly powerful means to guide the shadow removal and re-synthesis process. We use it to obtain two shadow masks, S_{src} and S_{tgt}, corresponding to the source and target sun directions respectively, by running a shadowcasting algorithm. If the geometry were perfect, these masks would tell the network precisely which pixels to brighten (resp. darken). However, because of errors in the stereo reconstruction, the masks typically contain significant artifacts and misalignments with respect to the actual shadows in the image.

While coarse masks are better than no shadow mask at all (see Section 5.5), we found that the success of the shadow removal procedure strongly depends on the quality of S_{src}. Similarly, the shadow re-synthesis suffers from errors in S_{tgt}. This led us to build an explicit *shadow refinement* step within our pipeline. We guide the refinement step by introducing *RGB shadow images*. These maps use color information from all images in the multi-view dataset to provide hints to the CNN on reconstruction inaccuracies.



(a) input views (b) calibrated cameras and 3D proxy

Fig. 3. (a) Our method takes as input a set of photos of an outdoor scene, shot from varying viewpoints (in this example 140). (b) We calibrate the cameras (shown in green) and build a 3D proxy of the scene using MVS. This reconstruction is approximate, as can be seen from the multiple holes (white) and erroneous over-reconstruction (e.g., blobs around palm trees with reconstructed sky). Our model learns to account for this uncertainty and generalizes well at test time.

Our overall model can thus be divided into three sub-components (Fig. 2). Two sub-networks independently refine the shadow masks S_{src} and S_{tgt}, and a third implements the final relighting given the illumination buffers and the refined shadow masks. The three components are trained jointly in an end-to-end, supervised fashion, using a training set of synthetic scenes. Our dataset contains ground truth source/target images, and approximate/ground truth shadow mask pairs.

3.1 Overall architecture

At a high-level our network is the composition of three sub-networks, two for the source (resp. target) shadow refinement tasks and one for relighting (Fig. 2). The refinement networks both take the RGB shadow images (Section 3.2.1) and the input images and predict refined greyscale shadow masks. These two refined shadow masks, along with the illumination buffers, are sent to the relighting sub-network which infers the target sun condition image and an overcast image. This 3-step approach is supported by recent results (e.g., [Wang et al. 2018]) showing that decomposing shadow detection and removal in two consecutive subtasks within the same network greatly improves quality. The overall architecture of our network is shown in Fig. 2; we use a ResNet [He et al. 2016; Johnson et al. 2016] for the shadow refinement and the relighting modules [Zhu et al. 2017]. We also experimented with a Unet-like architecture [Isola et al. 2017], that gave marginally inferior results. Our network outputs two images: the relit target image, and a “cloudy" rendering which we use to produce different degrees of overcast lighting conditions (Section 5.6.1).

3.2 Shadow refinement with RGB shadow images

Strong shadow cues are central to the shadow removal and re-synthesis process (see Section 5.5 for a comparison). The proxy

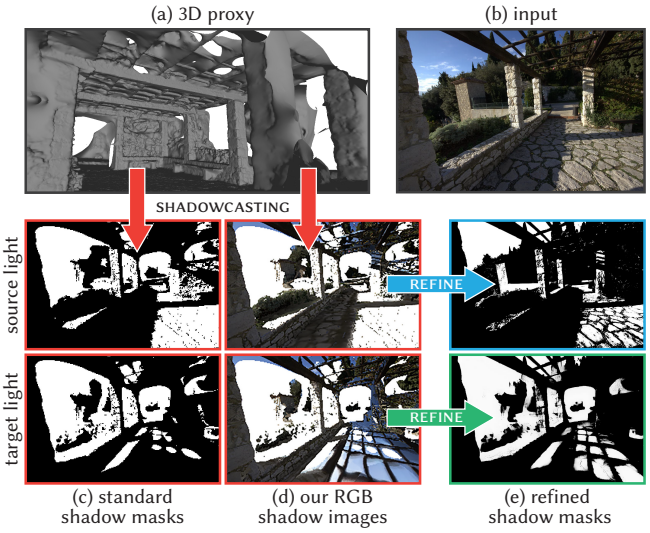


Fig. 4. We use the 3D proxy (a) to cast shadow masks corresponding to the *source* and *target* lighting conditions. Traditional shadow masks (c) already provide strong cues for a relighting model, but they often suffer from errors in the multi-view reconstruction. Our new RGB shadow images (d) are more expressive and help us recover from the proxy's errors. We process them with two independent shadow-refinement subnetworks to obtain finer shadow masks (e). In turn, these refined masks guide the removal of shadows in the input (b), and the synthesis of detailed cast shadows for the new lighting condition.

can be used to compute standard (greyscale) *shadow masks* (Fig. 4 (c), black pixels are occluded by the geometry). We found however that, because the proxy is only approximate, the shadow masks are usually too coarse, which motivates our shadow refinement pipeline. To reap the most benefits from this refinement, we introduce a novel representation — RGB shadow images — that is robust to inaccurate geometry (Section 3.2.2).

3.2.1 Independent source and target shadow refinement. Both source and target shadow maps are obtained from the proxy, and therefore need refinement (Fig. 4). We process the two maps *independently*, with two sub-networks that perform fundamentally different tasks.

Refining the source masks is an easier problem because the shadows in the input image are in exact correspondence with the shadow mask: the refinement network can use the image as guide.

This does not apply to the target masks. Since we want to change the lighting, the target masks are generally not aligned with the shadows in the input image, making the problem inherently more ambiguous. If instead, we used the *same* shared network for both tasks, the quality of the refined source shadows would degrade. Unlike specialized modules, a shared network cannot expect the masks to be consistently aligned with the image data.

We use synthetic data to create ground truth / proxy pairs for shadow refinement (Section 4.3). The source shadow refinement process uses the actual boundary in the input image, giving better overall results compared to the target shadow refinement (Fig. 4, (e)).

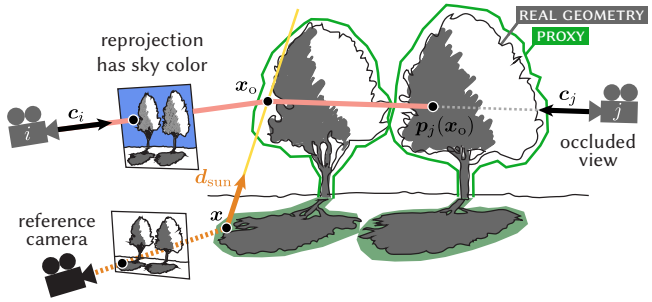


Fig. 5. Computation of the RGB shadow images. For a visible point $\boldsymbol{x}$ we reproject the shadow caster point $\boldsymbol{x}_o$ on the proxy into the input images. In this example, image i contributes a (blue) sky color, indicating the proxy is inaccurate at $\boldsymbol{x}_o$. The contribution of image j is reduced thanks to our weighting term because $\boldsymbol{x}_o$ is occluded by the rightmost tree from the point of view of camera j.

3.2.2 RGB shadow images. We introduce *RGB shadow images* to guide the refinement of both source and target shadow masks: this is a key element to the success of our solution. RGB shadow images S^{rgb} (Fig. 4, (d)) are created by reprojecting colors from all the other images in the multiview input. Their purpose is to help the network recover from over-reconstruction errors, e.g., the beams of the pergola in Fig. 4 appear connected as a solid ceiling. Our RGB shadow images will show blue pixels in the (incorrectly) shaded area corresponding to the sky, which easily disambiguates this error (compare (c) and (d) in Fig. 4).

We illustrate the computation of RGB shadow images in Fig. 5. For each pixel in shadow, we cast a ray in the direction of the sun $\boldsymbol{d}_{sun}$ from the corresponding 3D scene point $\boldsymbol{x} \in \mathbb{R}^3$ (see Fig. 5). The ray intersects the occluding proxy geometry at a point $\boldsymbol{x}_o$, that we reproject into the other input images. We accumulate a weighted average color collected from the other views. The weight for the contribution of a given image i to the color of a pixel in the RGB shadow image is computed as:

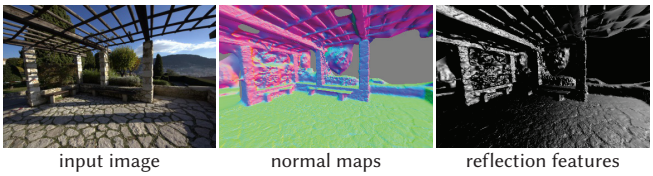
$$\frac{1}{||\boldsymbol{x}_o - \boldsymbol{p}_i(\boldsymbol{x}_o)||_2^2 \cdot |1 + \boldsymbol{c}_i^T \boldsymbol{d}_{sun}|^2 + \epsilon}, \quad (1)$$

where $\boldsymbol{c}_i$ is a unit vector giving the direction from camera i to $\boldsymbol{x}_o$, $\boldsymbol{p}_i(\boldsymbol{x}_o) \in \mathbb{R}^3$ is the first intersection of the camera ray defined by $\boldsymbol{c}_i$ with the proxy (Fig. 5) and $\epsilon = 1e{-}5$. The first term reduces contributions of images i when an object occludes $\boldsymbol{x}_o$ from the point of view of camera i. The second term tries to reduce reprojection error due to depth inaccuracy. It encodes a preference for views that are closer to the sun direction, in a similar spirit to blending weights for IBR [Buehler et al. 2001].

In addition to the weighted average of reprojected colors, we also maintain two additional pieces of information. First, we store the ratio of the distance from the visible point $\boldsymbol{x}$ to $\boldsymbol{x}_o$ to the distance from the current camera to $\boldsymbol{x}$; this provides a hint to the network on how soft the shadows should be. Second, we store the uncertainty of reconstruction, provided by the MVS algorithm since geometry is more likely to be erroneous whenever the algorithm's confidence is low. Our RGB shadow images can be computed quickly at test time for a captured scene.

3.3 Image-space geometric information via illumination buffers

The network takes as input a *source* and *target* lighting condition, that are defined by the respective sun positions. To help the network perform the lighting transformation, the first illumination buffers we provide are sun elevation for both source and target as scalars proportional to the angle between the horizon and the sun direction, as well as the sun directions in camera space as unit vectors. We also help the network determine surface lighting depending on sun orientation, by using the proxy to compute normal maps in camera space. Finally, we provide a reflection buffer that is the dot product between the direction from the camera to the surface and the mirror reflection of the incoming sun ray at the surface. These help the network synthesize illumination consistently and also affects shadow removal. Our illumination buffers are illustrated below, and their impact on final relighting quality is evaluated in Section 5.5.



input image normal maps reflection features

3.4 Training the model

The three sub-modules of our network are trained jointly in a supervised manner to minimize the sum of three losses:

$$\mathcal{L} = \mathcal{L}_{\text{relight}} + \mathcal{L}_{\text{src}} + \mathcal{L}_{\text{tgt}}. \tag{2}$$

These loss functions compare the accuracy of our network's predictions (the final relit image as well as both intermediate refined shadow masks) to synthetic ground truth, which we detail in Section 4.

To refine the source shadow mask, a straighforward L_1 loss proved sufficient. Intuitively, this task is less ambiguous than refining the target shadow maps because the input image contains the source shadows:

$$\mathcal{L}_{\text{src}} = \mathbb{E}\left[\left|r_{\text{src}}(S^{\text{rgb}}_{\text{src}}, I) - S^{\star}_{\text{src}}\right|\right], \tag{3}$$

where r_{src} is the source refinement network, $S^{\star}_{\text{src}}$ is the ground truth shadow mask, I is the input image, and $S^{\text{rgb}}_{\text{src}}$ is the source RGB shadow image. The operator $\mathbb{E}$ denotes expectations taken over the training set.

For the target shadow refinement however, the network has less information to exploit from the input image, so we use a more complex perceptual loss:

$$\mathcal{L}_{\text{tgt}} = \mathbb{E}\left[w_1 \cdot \mathcal{P}(r_{\text{tgt}}(S^{\text{rgb}}_{\text{tgt}}, I), S^{\star}_{\text{tgt}})\right], \tag{4}$$

with r_{tgt} the target refinement network, $S^{\text{rgb}}_{\text{tgt}}$ the target RGB shadow image, $S^{\star}_{\text{tgt}}$ the ground truth target shadow mask, and w_1 a weight map. $\mathcal{P}$ is a perceptual loss function. It extracts features from the two images independently using a pretrained VGG19 network and compares them with an L_1 loss. We use the implementation of Chen & Koltun [2017].

In practice, pixels shadowed by geometry that is not reconstructed in the proxy are fundamentally ambiguous. They tend to bias the target refinement network towards conservative outputs (see Section 4.3). We reduce the contribution of these pixels using a binary mask computed from the ground truth shadow mask. We set $w_1 = \frac{1}{10}$ for the masked pixels, and 1 otherwise (value found empirically to give satisfactory results).

For the relighting network, we also use a weighted perceptual loss. We weight the loss using the difference between the ground truth and proxy shadow image so that we do not penalize parts of the shadow mask where the refinement step failed. Specifically, the weight is given by $w_2 = 1 - 0.9|r_{\text{tgt}}(S^{\text{rgb}}_{\text{tgt}}, I) - S^{\star}_{\text{tgt}}|$.

The overall goal of the relighting network is to produce a relit image I^{R}, which we encourage to match the ground truth target lighting condition $I^{\star}$ using, again, an image-space perceptual criterion: .

$$\mathcal{L}_{\text{relight}} = \mathbb{E}\left[w_2 \cdot \mathcal{P}(I^{\text{R}}, I^{\star})\right]. \tag{5}$$

Note that I^{R} depends on the input image, the illumination buffers and the *refined* source and target shadow masks.

3.4.1 Details. The weights of all the convolutional layers are initialized according to He et al.'s recommendation [He et al. 2015] and the biases to 0. We optimize the network parameters using the ADAM solver [Kingma and Ba 2015], we train with a batch size of 4, and a learning rate of 2×10^{-4}. The remaining parameters of the ADAM optimizer are kept to the values recommended by the authors. Our model is implemented in Tensorflow [Abadi et al. 2015]. Unless specified otherwise, the models were trained on a NVIDIA GTX 1080 Ti GPU until the loss stopped improving (typically 3–4 days). The architecture is a 64-channel ResNet for the relighting module and a 16-channel ResNet for the shadow refiners, following [Zhu et al. 2017]

4 SYNTHESIZING TRAINING DATA

Capturing a large-scale dataset of real photographs to train our multi-view relighting network would be cumbersome and fraught with practical difficulties. To guarantee sufficient coverage of the lighting scenarios, such a campaign would have to cover many different locations, maintain strictly fixed viewpoints during capture, and require day-long (or even month-long) capture sessions with many cameras. Even if this approach were practically possible, it would lack in diversity, e.g., for the kind of lighting conditions and scene content available. In addition, data for shadow refinement supervision cannot be directly captured.

To bypass these issues, we use synthetic training data and render photo-realistic images using the Mitsuba [Jakob 2010] pathtracer. We gathered a set of 10 synthetic scenes from which we compute the data required for training. This approach allows us to generate arbitrary lighting conditions easily and have full control over the supervision at training time. To maximize diversity while keeping rendering time under control, we factorize the lighting computation by separately rendering the sun and sky contributions and compositing the two at training time. The use of synthetic data also allows us to render ground truth shadow masks $S^{\star}_{\text{tgt}}, S^{\star}_{\text{src}}$, which is critical to our shadow refinement sub-network (Section 3.2).

Fig. 6. A sample viewpoint from each of our 10 ground truth training scenes.

The key requirement for our training images is that they closely resemble real photographs. That is, the scenes must contain highly detailed models of outdoors scenes, with realistic materials. For this reason, we chose to use professionally built models (either purchased or freely available) and develop a set of data augmentation techniques. Our experiments show that, even though our network is trained entirely on this synthetic dataset, it generalizes well to real images (Fig. 13, 16).

4.1 Synthetic scenes

We gathered 10 different outdoor 3D scenes to generate our training data; a sample viewpoint from each scene is shown in Fig. 6. The first 8 scenes were professionally modeled scenes we purchased[1]. We also used the large scene published by NVIDIA[2] and created two separate subscenes (a street and a square). The scenes are in standard industry formats (typically Autodesk 3DSMax), and include hand-crafted materials with complex shading trees which we export as Mitsuba scene description files [Jakob 2010].

We render the scenes using path tracing and Mitsuba's physically-based sun model, with HDR sky environment maps from Stumpgel et al. [2004]. We remove the sun from these environment maps by mirroring the envmap. The physically-based model provides correct colors for the sun at different sun elevations, as well as a sky environment map [Hosek and Wilkie 2012]. We apply the average color and intensity shift of the sky for a given sun position during compositing (Section 4.2).

4.2 Photo-realistic rendering, layer decomposition and compositing-based data augmentation

Path-tracing complex outdoors scenes with a physically-based sun/sky model is expensive: rendering a converged image at 1024×768 takes about 10 minutes on our 400-core cluster. However, we noticed that our method works well with relatively noisy images, so we use 64 samples per pixel for all our renderings, with good results. This corresponds to a recent observation that learning with noisy rendering data can be robust [Lehtinen et al. 2018]. Generating the same resolution image with these settings takes about 10 seconds on the same cluster. For our dataset which contains about 17,000 rendered images this approach reduces rendering time from about 100 days to only 2 days.

For each training scene we select around 30 different viewpoints to obtain as much content variety as possible. To increase the number of lighting conditions within a fixed rendering budget, we render sky and sun illumination as two separate images that we composite on-the-fly at training time. This allows us to apply random intensity variations before we generate the final image. For each scene and each viewpoint , we render 49 sun positions, and 5 sky conditions, varying the cloud coverage. We store these render buffers as floating point linear images. Thus, for each viewpoint we have $5 * 49 = 245$ lighting conditions, before applying any intensity data augmentation. This leads to $245^2 = 60$K pairs of training lighting conditions per viewpoint.

On-line compositing. For a given training step, we need to generate a source "input" image, corresponding to the input photo we will use at test time and a target ground truth image, corresponding to the desired image relit with the target lighting configuration.

We start by randomly selecting 2 out of the 49 sun position images to be the source and target conditions and we randomly select a single sky condition image used for both, scaling the sky with the corresponding average color shift computed using the sky model [Hosek and Wilkie 2012]. Sky and sun illumination are highly correlated so this is not strictly physically accurate, but the quality of the results was satisfactory despite this approximation. We also randomly scale the sun intensity separately per channel, and randomly scale all channels of sky intensity.

Data-augmentation. We first randomly scale our images and select a random crop of 256×256 pixels. Real-world images have a variety of exposure and white balance settings. To be robust to this variety in the input, we apply random variations to both source and target images during training.

We next take the source and target linear images with all the random perturbations applied, and perform gamma correction, with

[1]Scenes purchased from http://evermotion.org, taken for collections Archexteriors vol. 17 and 22.
[2]https://developer.nvidia.com/orca/amazon-lumberyard-bistro

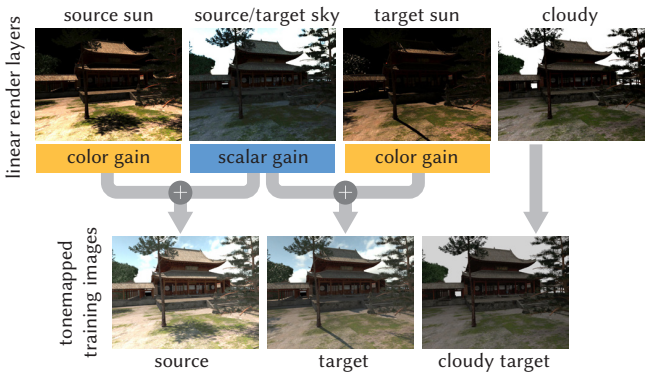


Fig. 7. **Top**: linear images from left to right, source sun image, sky image, target sun image, cloudy image. **Bottom**: composited source, composited target and cloudy images, after data augmentation and tone-mapping. All linear images overexposed for visibility.

small random variation on the gamma value. We use the same random variables for all transformations, including compositing, for source and target to preserve coherence; these are also applied to the RGB shadow images. We illustrate this process in Fig. 7; details of all the processing steps for compositing and augmentation are provided in the supplemental. Our data augmentation scheme is critical to the performance of our algorithm, as we show in an ablation test (Section 5.5).

In addition to the relit image, we train our network to produce a "cloudy" layer, i.e., an image lit only by a uniform mid-gray sky. We use it to approximate different degrees of overcast conditions.

4.3 MVS reconstruction of synthetic ground truth scenes

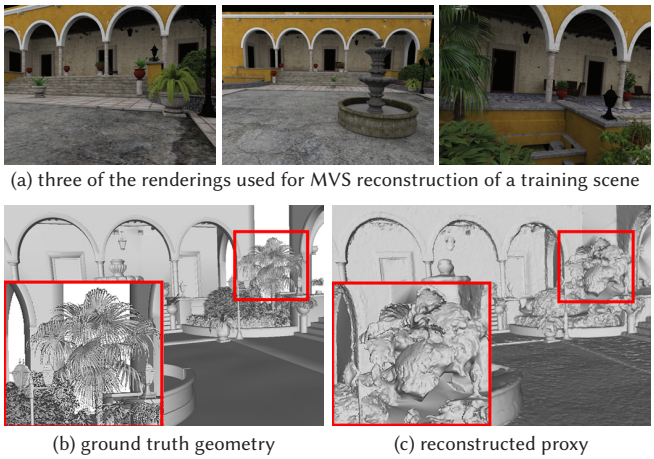
When relighting a real scene, we only have the approximate proxy representation of the scene to generate shadow images. For shadow refinement to be successful at test time, the network needs to learn the mapping between approximate proxy shadows and ground truth shadows at training time. To achieve this we create a second representation of each synthetic scene by rendering a set of views, subsequently used as if they were photographs of a real scene. These photos are then processed with SfM and MVS to create a proxy of the synthetic scene, with the typical reconstruction artifacts of this process (Fig. 8). For more details on this step, please see supplemental materials.

4.4 Training with synthetic data

We train our network using both representations of each synthetic scene. The ground truth geometry and materials are used to render the sun and sky layers, and to create the ground truth greyscale shadow masks. The proxy is used to generate the illumination buffers and RGB shadow images. Details of the RGB shadow image generation for training are given in supplemental material.

5 RESULTS AND EXPERIMENTS

We have implemented our method in both interactive and batch processing contexts. To perform relighting, we require a set of calibrated cameras and a proxy. The user must then specify the source



(a) three of the renderings used for MVS reconstruction of a training scene

(b) ground truth geometry (c) reconstructed proxy

Fig. 8. To bridge the gap between our synthetic training data and real multi-view datasets, we purposely degrade the quality of the training geometry by running a multi-view stereo algorithms on our renderings (a). Compared to the ground truth geometry (b), the proxy (c) is inaccurate and misses many details (e.g., the trees).

sun position by clicking on a shadow caster and the corresponding shadow on the textured mesh (see supplemental video). We present quantitative and qualitative results, comparisons to previous work, ablation studies and applications. Our results and video can be found at http://fungraph.inria.fr/deep-relighting.html.

5.1 Qualitative results

We show the output of our relighting method for a variety of scenes, under a large range of lighting conditions in Fig. 16. Our method successfully removes and resynthesizes shadows, and achieves convincing changes in illumination levels for different times of day and lighting conditions. We present an extensive set of relighting results for the 8 different scenes in Fig. 16 with large sun arc movements in supplemental material. These include 2 drone video captures (first two rows of Fig. 16), 2 scenes from Duchene et al. [2015] (last two rows of Fig. 16) and 4 scenes we captured ourselves.

5.2 Comparison to a neural network baseline

Assuming proper training data is available, a natural approach to relighting outdoor scenes would be to train a standard model such as a ResNet [He et al. 2016; Johnson et al. 2016] to transform an input image given a target sun direction. We trained such a model with the image and source/target sun direction layers as input, using the same data augmentation (sky/sun rendering, exposure and white balance) as our approach. As shown in Fig. 9, the images it produces are not satisfactory. The network completely ignores the sun direction input layers and produces an image with reduced intensity and shadows that are only partially removed. It also does not synthesize cast shadows that are consistent with the target sun direction.

This purely image-based baseline simply does not have enough information to solve the severely ill-posed relighting problem. For instance, even with its large field of view, the ResNet cannot properly

(a) input (b) ResNet output (c) our output

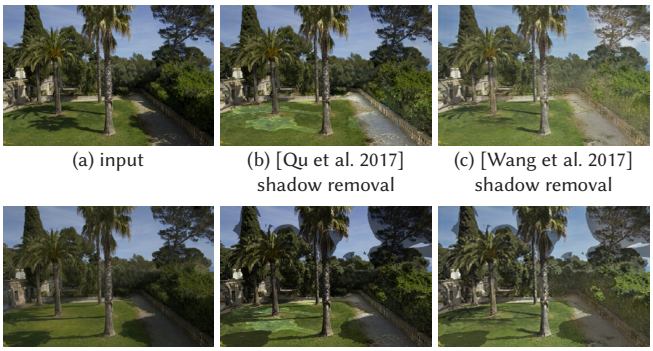
Fig. 9. Relighting is a challenging task for standard image-to-image networks. Even when provided with our auxiliary inputs and shadow masks, a ResNet model (b) struggles to remove cast shadows in the input (a), and cannot generate globally-consistent color changes and plausible novel shadows. Our model gives much more realistic results (c).

deal with the non-local nature of cast shadows. Furthermore, this baseline has no notion of surface appearance or surface orientation.

5.3 Comparison to previous work

As a second comparison we first apply a shadow removal algorithm ([Qu et al. 2017] or [Wang et al. 2018]), then cast a new shadow using the proxy geometry. Fig. 10 shows the shadow removal generally fails on our real test images. Also, the proxy is often too approximate to use its cast shadows directly, justifying our shadow refinement approach. It is important to note however that unlike our method, neither of these shadow-removal techniques uses multi-view information, and thus have much less information to work with than our approach.

We also compare to the relighting algorithm of Duchêne et al. [2015], where we have used the same 3D reconstruction as in their original method. We see that Duchêne et al. achieve good quality shadow casting close to the original sun direction, but the shadow shape is completely incorrect when moving further away. The method also suffers from residual artifacts due to incorrect shadow classification (examples highlighted by red squares). These artifacts can be better seen in the companion video.



(a) input (b) [Qu et al. 2017] shadow removal (c) [Wang et al. 2017] shadow removal

(d) our output (e) [Qu et al. 2017] + proxy cast shadows (f) [Wang et al. 2017] + proxy cast shadows

Fig. 10. We compare our method to a baseline that removes shadows using off-the-shelf algorithms then casts new shadows from our proxy geometry. (b) and (c) are the output of the shadow-removal algorithms from input (a). (e) and (f) show the same images with new shadows generated using the proxy. Our output is significantly cleaner (d). [Qu et al. 2017] [Wang et al. 2018]

(a) input (b) [Duchêne et al. 2015] (c) our output

Fig. 11. Duchêne et al. [2015] often leaves shadow residuals (b), bottom. Their method also breaks when the desired relighting is far from the input (b), top. Our method is more robust and can synthesize significant lighting changes (c). [Duchêne et al. 2015]

(a) input (b) ground truth (c) our output

Fig. 12. We evaluate our model on a held-out synthetic scene (a) for which we can generate arbitrary novel lighting conditions (b). Our model can faithfully predict the novel illumination (c) even though it has not been trained on this scene and has only access to the degraded geometry (proxy) and input images.

5.4 Comparison to synthetic and real ground truth

We show comparison to a synthetic scene held out from the training data in Fig. 12. Note that we used a reconstructed proxy and not the perfect, ground truth geometry to obtain these results.

We also show a qualitative ground truth comparison with a real scene, in Fig. 13. For this, we photographed the same scene, at different times of day, with the same viewpoint.

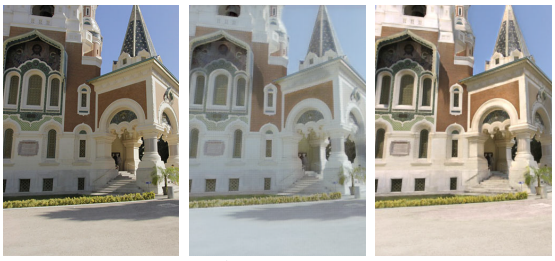
5.5 Model ablations

We performed several ablations of our model. For each analysis we trained the different configurations for 100 epochs. We held out one synthetic scene for testing and trained our network on the others. Table 1 summarizes the numerical error for the different ablations

We present interactive side-by-side comparisons of the different ablations as a web page in supplemental materials for three different scenes.

Data augmentation. Our data augmentation procedure randomizes exposure, white balance and gamma correction. It is critical

to the success of the network, especially in generating correct illumination levels. Below, we show an example output of our model trained without data augmentation.



input no data augmentation our output

Illumination features. When we remove the illumination features, the network has difficulty finding the correct illumination levels, and generates inconsistent results. These layers help the network alter the image intensity consistently, improving shadow removal:

input no illumination buffers our output

Shadow refinement. If we only remove shadow refinement from our solution, shadow removal is also worse, and shadow re-synthesis exhibits ghosting artifacts:

input no shadow refinement our output

RGB shadow images. If we deactivate our RGB shadow images and use standard gray-scale shadow masks instead, the network cannot overcome over-reconstruction artifacts and the resynthesized shadows mostly follow the masks:

input no RGB shadow images our output

Table 1. We evaluate the error of our relighting numerically on a held-out ground truth synthetic scene. We report the average L_1 and L_2 pixel error. The input illumination buffers, shadow refinement subnetworks and data augmentation procedure all contribute to the final quality of our result.

model	L_1 error	L_2 error
our full model	**0.131**	**1.98e−4**
no shadow refinement	0.179	2.43e−4
no RGB shadow images	0.184	2.72e−4
no illumination buffers	0.200	2.51e−4
no image augmentation	0.445	4.54e−4

5.6 Applications

Our method can be used in several different contexts. We present four potential applications: interactive relighting, drone-video relighting, relighting for image-based rendering and relighting of reconstructed textured meshes.

5.6.1 Interactive Relighting. In our interactive application the user selects the input image to relight, and the network produces the relit image together with the "cloudy" layer. We can simulate varying levels of overcast conditions by inserting a blurring kernel after shadow refinement and before relighting, providing a user-controlled "cloudiness" parameter. We then blend between the resulting relit and cloudy image to produce the output (Fig. 14).

We have developed an integrated interactive viewer by calling tensorflow with a CUDA/OpenGL coupling, allowing interactive performance.

Performance numbers: 5-8 frames per-seconds on an Dell Precision 7810 computer with an NVIDIA 1080GTX GPU at 1080p resolution (see video).

5.6.2 Drone video relighting. We extract frames from a drone video and perform standard multi-view reconstruction. We can then individually relight the frames of the video using our approach, either at a single different time or dynamically changing the lighting during the video (Fig. 1, top row). This is best seen in the supplemental video. Our algorithm treats each frame independently, without explicit temporal regularization, so we sometimes observe flickering in the rendered videos. This is easily corrected using a post-processing temporal smoothing method like that of Lai et al. [2018].

5.6.3 Relighting for Image-Based Rendering (IBR). We have integrated our relighting in an interactive IBR system implementing [Buehler et al. 2001], by relighting the blended novel view on-the-fly. The ability to relight for IBR overcomes one of the major limitations of these techniques that are otherwise restricted to the lighting conditions of capture. Please see the video for examples.

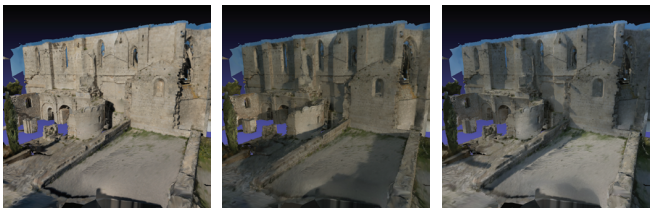
5.6.4 Relighting for Reconstructed Textured Meshes. We can relight all the images for a given multi-view dataset in a new sun position, and then re-run the final texturing step after geometric reconstruction. In supplemental, we provide three meshes with different versions of the same scene, i.e., two conditions in addition to the original captured lighting (Fig. 15).

(a) input photo (b) same scene, 3h30 later (real photo) (c) our output

Fig. 13. Our network generalizes to real input images (a) and produces photorealistic outputs (c) that closely match real, novel lighting conditions (b).

input 50% cloudiness 100% cloudiness

Fig. 14. Our model exposes a user-controllable "cloudiness" parameter to modulate between *sunny* and *overcast* conditions.



(a) original mesh (b) relit mesh 1 (c) relit mesh 2

Fig. 15. Our algorithm can also be used to relight an input textured mesh (a) to different lighting conditions (b), (c).

5.7 Performance breakdown

Our pipeline takes less than 10 minutes from the begining for the multi-view calibration procedure to the final relit result. In particular our neural network runs at interactive rates, which enables a user to alter the lighting dynamically. We report the computational cost for a typical scene with 109 input photos in the table below.

preprocess	camera calibration	1 min
	proxy reconstruction	6 min
	manual sun position input	15 s
runtime	rendering RGB shadow images	40 ms
	rendering other buffers	<10 ms
	network inference	70 ms

6 DISCUSSION AND CONCLUSION

Limitations. Our method generally produces plausible results for the scenes we tested, including scenes from previous work, drone captures, internet image datasets and our own captures. Occasionally, slight shadow residue is visible in some views of a given dataset (see Fig. 17(a)); this typically occurs in overexposed or very dark image regions where shadow information is unreliable. Our network may occasionally produce small "checkerboard" artifacts, that come from the "deconvolution" upsampling layer. This is a common issue with this type of network.

Our goal is plausibility, and therefore in most cases the network does not hallucinate *additional* shadows when no occluder geometry is available. This can be seen in Fig. 17(b) where the top branches of the palm tree are missing from the relit shadow at the input sun position. However the result is plausible since the original shadow is cleanly removed.

6.1 Conclusion

We present a learning method guided by approximate reconstructed geometry for multi-view relighting. Important elements of our relighting solution are the shadow refinement subnetworks, guided by our newly introduced RGB shadow images, as well as illumination buffers. The use of synthetic data allows generation of highly diverse ground truth data, and the creation of a proxy representation in addition to ground truth geometry for each synthetic scene allows supervised training for shadow refinement.

Our results show that by performing relighting of multi-view datasets we greatly increase their utility for traditional applications such as photogrammetry meshing and IBR, but we also demonstrate very powerful novel image and video manipulation applications for drone footage and photos of landmarks, where internet-based multi-view data is available.

ACKNOWLEDGMENTS

The authors thank G. Kopanas, L. Boiron and S. Morgenthaler for the development of the 3DSMax to Mitsuba exporter and ground truth rendering system. Thanks to A. Bousseau and K. Sunkavalli for proofreading earlier drafts. Funding was provided by the European Commission grants EMOTIVE H2020 project No. 727188 (https://emotiveproject.eu/) and ERC Advanced Grant FUNGRAPH No. 788065 (http://fungraph.inria.fr). Drone footage copyright Drones Yucatán (Fig. 16, 1st row) and Namyeska - info@namyeska.com (Fig. 1 and 16, 2nd row). We thank T. Hilfer for his photo (Fig. 1, 2nd row).

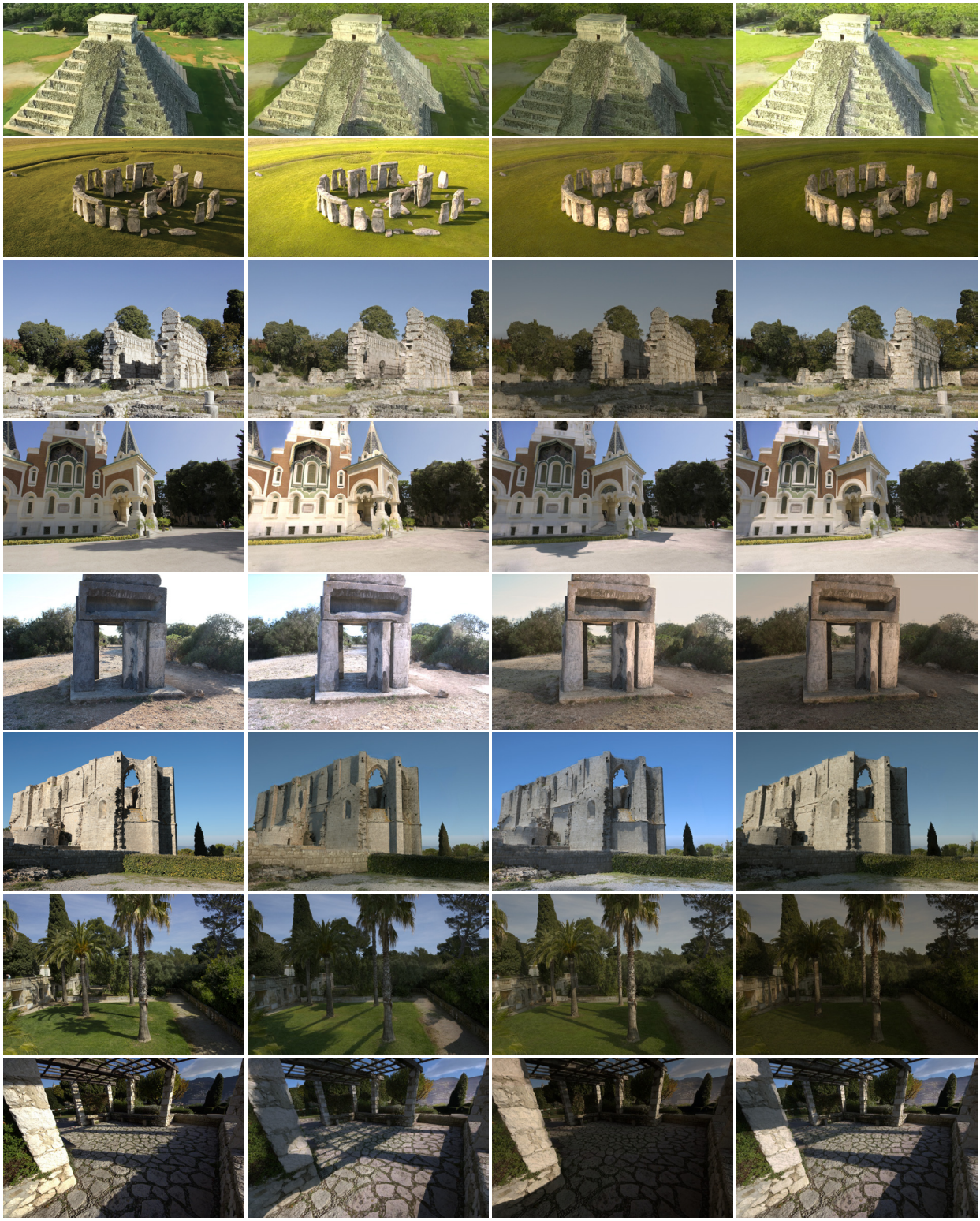
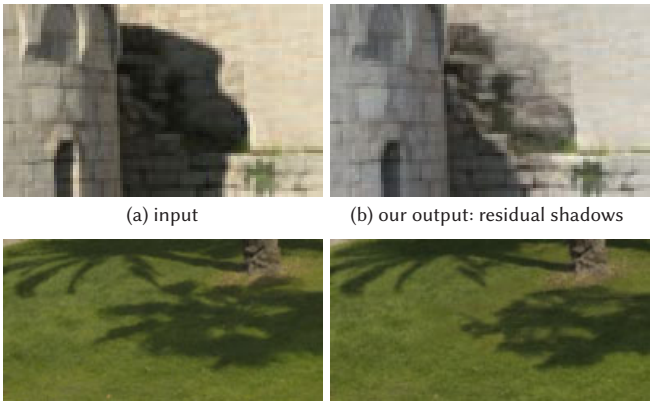


Fig. 16. Results using our relighiting network. The leftmost column is the input, followed by three outputs corresponding to different sun positions. First and second row respectively generated using the Chichen Itza drone video (copyright Drones Yucatán youtu.be/qkveKd3nW9w) and Stonhenge drone video (copyright Namyeska youtu.be/JHeDP7_YBos) both used with permission.



(a) input (b) our output: residual shadows

(c) ground truth (d) our output: missing geometry

Fig. 17. Thanks to our RGB shadow images, our algorithm can generally refine inacurrate shadows. However it sometimes confuses texture detail with the input shadows (a), which creates a visible shadow residual (b). When a scene object is not properly reconstructed by MVS (shadow of the palm leaves in (c)), our model cannot hallucinate the missing shadows (d).

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proceedings SIGGRAPH*.

Qifeng Chen and Vladlen Koltun. 2017. Photographic Image Synthesis with Cascaded Refinement Networks. *CoRR* abs/1707.09405 (2017). arXiv:1707.09405 http://arxiv.org/abs/1707.09405

Paul Debevec. 2002. Image-based lighting. *IEEE Computer Graphics and Applications* 22, 2 (2002), 26–34.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 145–156.

Sylvain Duchêne, Clement Riant, Gaurav Chaurasia, Jorge Lopez-Moreno, Pierre-Yves Laffont, Stefan Popov, Adrien Bousseau, and George Drettakis. 2015. Multi-View Intrinsic Images of Outdoors Scenes with an Application to Relighting. *ACM Transactions on Graphics (TOG)* 34, 5 (Nov. 2015).

Graham D Finlayson, Steven D Hordley, Cheng Lu, and Mark S Drew. 2006. On the removal of shadows from images. *IEEE transactions on pattern analysis and machine intelligence* 28, 1 (2006), 59–68.

Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. 2007. Multi-view stereo for community photo collections. In *International Conference on Computer Vision (ICCV)*.

Maciej Gryka, Michael Terry, and Gabriel J. Brostow. 2015. Learning to Remove Soft Shadows. *ACM Transactions on Graphics (TOG)* 34, 5 (Oct. 2015).

Ruiqi Guo, Qieyun Dai, and Derek Hoiem. 2011. Single-image shadow detection and removal using paired regions. In *CVPR, 2011*. IEEE, 2033–2040.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *CoRR* (2015).

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. 2017. Deep outdoor illumination estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 2.

Lukas Hosek and Alexander Wilkie. 2012. An analytic model for full spectral sky-dome radiance. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 95.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*. Springer, 694–711.

Yoshihiro Kanamori and Yuki Endo. 2018. Relighting humans: occlusion-aware inverse rendering for fullbody human images. *ACM Transactions on Graphics (TOG)* 37, 270 (2018), 1–270.

Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. 2011. Rendering Synthetic Objects into Legacy Photographs. *ACM Transactions on Graphics (TOG)* 30, 6, Article 157 (Dec. 2011), 12 pages.

Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 2014. 3D object manipulation in a single photograph using stock 3D models. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 127.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR* (2015).

Johannes Kopf, Boris Neubert, Billy Chen, Michael Cohen, Daniel Cohen-Or, Oliver Deussen, Matt Uyttendaele, and Dani Lischinski. 2008. *Deep photo: Model-based photograph enhancement and viewing*. Vol. 27. ACM.

Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, and George Drettakis. 2012. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics (TOG)* 31, 6 (2012).

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning Blind Video Temporal Consistency. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 179–195.

Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. 2009a. Estimating natural illumination from a single outdoor image. In *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 183–190.

Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. 2009b. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 131.

Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. 2010. Detecting ground shadows in outdoor consumer photographs. In *European conference on computer vision*. Springer, 322–335.

Edwin H Land and John J McCann. 1971. Lightness and retinex theory. *Josa* 61, 1 (1971), 1–11.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. 2018. Noise2Noise: Learning Image Restoration without Clean Data. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2971–2980.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*. 700–708.

Céline Loscos, Marie-Claude Frasson, George Drettakis, Bruce Walter, Xavier Granier, and Pierre Poulin. 1999. Interactive virtual relighting and remodeling of real scenes. In *Rendering Techniques 99*. Springer, 329–340.

Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep photo style transfer. *CoRR, abs/1703.07511* 2 (2017).

Stephen R Marschner and Donald P Greenberg. 1997. Inverse lighting for photography. In *Color and Imaging Conference*, Vol. 1997. Society for Imaging Science and Technology, 262–265.

Vincent Masselus, Pieter Peers, Philip Dutré, and Yves D Willems. 2003. Relighting with 4D incident light fields. *ACM Transactions on Graphics (TOG)* 22, 3 (2003), 613–620.

Vincent Masselus, Pieter Peers, Philip Dutré, and Yves D. Willems. 2004. Smooth reconstruction and compact representation of reflectance functions for image-based relighting. In *Rendering Techniques 2004*. Eurographics Association, Norrkoping, Sweden, 287–298.

Ankit Mohan, Jack Tumblin, and Prasun Choudhury. 2007. Editing soft shadows in a digital photograph. *IEEE Computer Graphics and Applications* 27, 2 (2007).

Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. 2007. Post-production facial performance relighting using reflectance transfer. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 52.

Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, and Rynson W. H. Lau. 2017. DeshadowNet: A Multi-Context Embedding Deep Network for Shadow Removal. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

CapturingReality RealityCapture. 2016. RealityCapture. \protect{http://https://www.capturingreality.com/}

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-assisted Intervention (MICCAI)*. Springer, 234–241.

Andres Sanin, Conrad Sanderson, and Brian C Lovell. 2012. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern recognition* 45, 4 (2012), 1684–1695.

Yichang Shih, Sylvain Paris, Frédo Durand, and William T Freeman. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 200.

Yael Shor and Dani Lischinski. 2008. The shadow meets the mask: Pyramid-based shadow removal. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 577–586.

Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. 2017. Neural face editing with intrinsic image disentangling. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 5444–5453.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo Tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics (TOG)* 25, 3 (July 2006), 835–846.

Jessi Stumpfel, Chris Tchou, Andrew Jones, Tim Hawkins, Andreas Wenger, and Paul Debevec. 2004. Direct HDR capture of the sun and sky. In *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. ACM, 145–149.

Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. 2007. Factored time-lapse video. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 101.

Chris Tchou, Jessi Stumpfel, Per Einarsson, Marcos Fajardo, and Paul Debevec. 2004. Unlighting the parthenon. In *ACM Siggraph 2004 Sketches*. ACM, 80.

Jifeng Wang, Xiang Li, and Jian Yang. 2018. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1788–1797.

Yang Wang, Lei Zhang, Zicheng Liu, Gang Hua, Zhen Wen, Zhengyou Zhang, and Dimitris Samaras. 2009. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 11 (2009), 1968–1984.

Yair Weiss. 2001. Deriving intrinsic images from image sequences. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 2. IEEE, 68–75.

Zhen Wen, Zicheng Liu, and Thomas S. Huang. 2003. Face Relighting with Radiance Environment Maps. In *CVPR*.

Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. 2005. Performance relighting and reflectance transformation with time-multiplexed illumination. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 756–764.

Tai-Pang Wu, Chi-Keung Tang, Michael S Brown, and Heung-Yeung Shum. 2007. Natural shadow matting. *ACM Transactions on Graphics (TOG)* 26, 2 (2007), 8.

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. 2018. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 126.

Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. 1999. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 215–224.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.